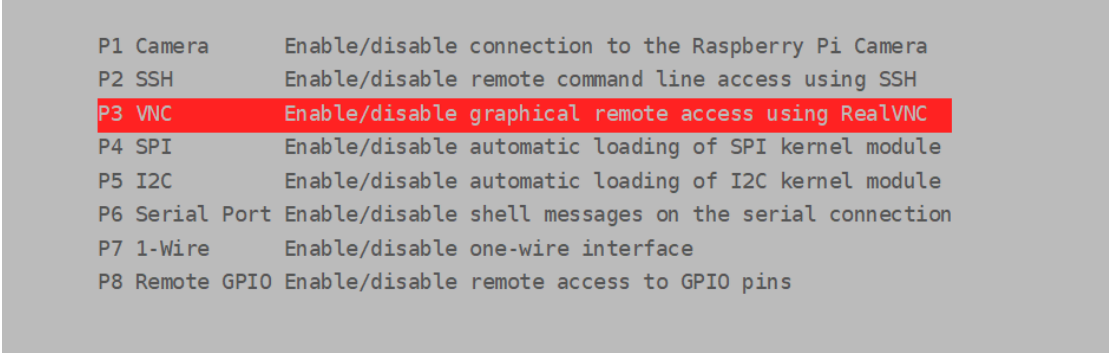


系统装好之后 利用 Putty 或者是其他工具登录 我们这里采用 FinalShell 来进行登录，使用 SSH 方式，需要在根目录下面创建一个 ssh 文件。

需要先配置显示器，不然没法进入到界面当中查看显示。

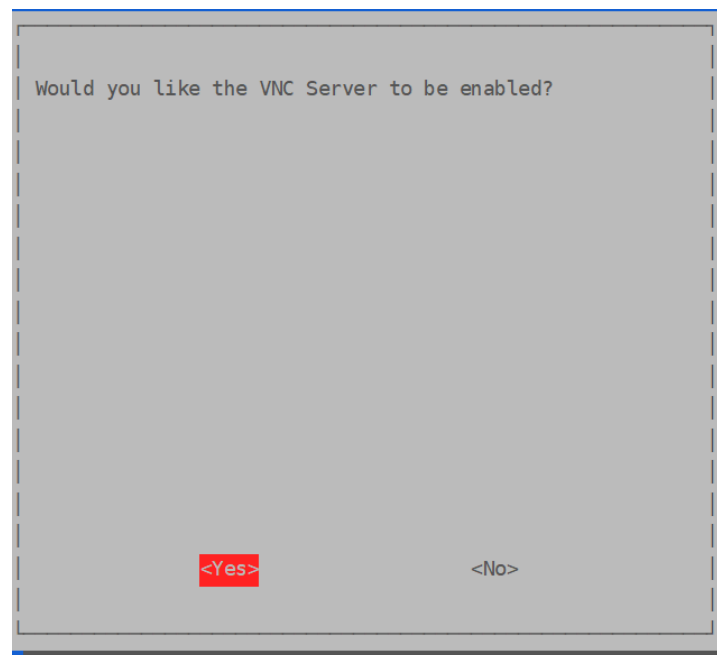
`sudo raspi-config`

这个是配置显示屏

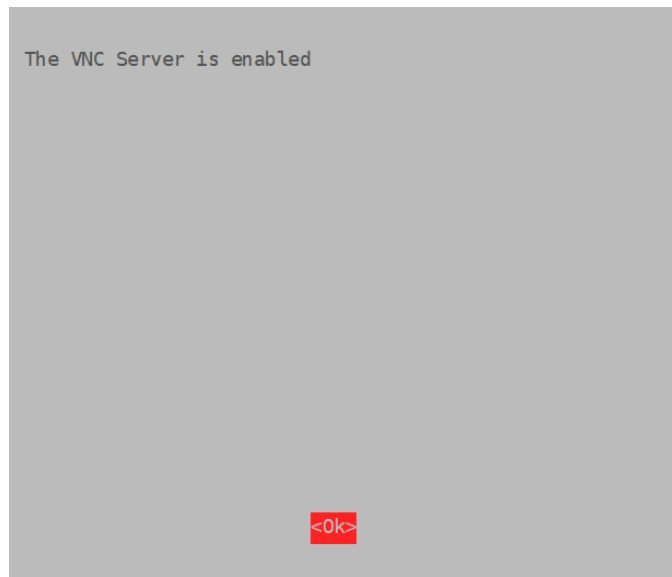


```
P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins
```

配置 VNC 开启，初次的时候，应该要装一些驱动，如果没有会提示安装的。



点击 yes 进入安装



被我们打开了



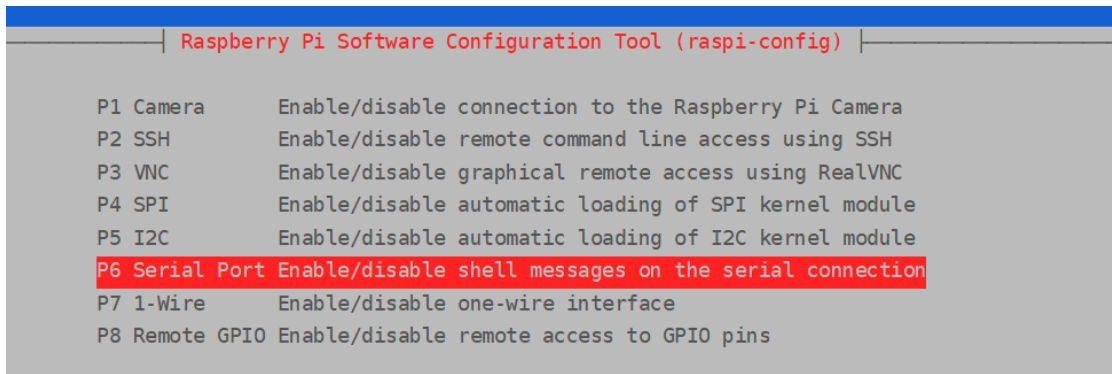
提示重启，表明 OK,分辨率一定要设置，否则不能用

ls /dev/ttyS*

下面查看 port 端口情况，默认 ttyS0 是 minicom。就是板子上的 GPIO 的串口

```
pi@raspberrypi:~$ ls /dev/ttyS*  
ls: cannot access '/dev/ttyS*': No such file or directory
```

没有串口，我们要使能串口



设置好，重启生效的，这个是个坑，否则一直不行

```
pi@raspberrypi:~$ ls -l /dev |grep ttyUSB
pi@raspberrypi:~$ ls /dev/ttyS*
/dev/ttyS0
```

查到设备端口了。

由于树莓派自带的服务器源在国外，更新文件非常慢，所以我们可以更新过来。

`sudo nano /etc/apt/sources.list`

输入以上指令

将下面 2 句话加入，他自带的可以屏蔽掉

```
GNU nano 3.2 /etc/apt/sources.list

#deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
deb http://mirrors.aliyun.com/raspbian/raspbian/ buster main non-free contrib
deb-src http://mirrors.aliyun.com/raspbian/raspbian/ buster main non-free contrib
```

`sudo apt-get install minicom`

安装下 minicom

如果安装失败，设置下 DNS，

`sudo nano /etc/resolv.conf`

修改 DNS：

nameserver 8.8.8.8

nameserver 8.8.4.4

修改 DNS :

nameserver 8.8.8.8

nameserver 8.8.4.4

sudo apt-get update

```
pi@raspberrypi:~ $ sudo apt-get install minicom
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  lrzsz
The following NEW packages will be installed:
  lrzsz minicom
0 upgraded, 2 newly installed, 0 to remove and 112 not upgraded.
Need to get 329 kB of archives.
After this operation, 1,332 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian buster/main armhf lrzsz armhf 0.12.21-10 [79.8 kB]
Get:2 http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian buster/main armhf minicom armhf 2.7.1-1 [250 kB]
Fetched 329 kB in 1s (249 kB/s)
Selecting previously unselected package lrzsz.
(Reading database ... 165829 files and directories currently installed.)
Preparing to unpack .../lrzsz_0.12.21-10_armhf.deb ...
```

设置好重启

安装驱动

先安装 sudo apt-get install raspberrypi-kernel-headers

安装一下这个内核编译的软件，不然不能编译，如果已经安装了，就不需要了。

export ARCH=arm 因为树莓派是 ARM,内核的，所以要设置下内核编译选项

压缩文件用 ZIP 的，这样解压出来的文件内容不会丢失。

```
pi@raspberrypi:~ $ tar -xvf Quectel_Linux_USB_Serial_Option_Driver_20200720.tgz
20200720/
```

这个指令就是将驱动文件进行解压。

tar -xvf Quectel_Linux_USB_Serial_Option_Driver_20200720.tgz

将文件解压下。然后进入到对应的路径里面

```
pi@raspberrypi:~ $ uname -r
5.4.79-v7l+
```

获取到我们的 Linux 版本内容，然后到资料里面找对应的驱动进行使用

v5.5.1 找到这个文件作为驱动编译使用

```
pi@raspberrypi:~ $ cd 20200720
pi@raspberrypi:~/20200720 $ cd v5.5.1
pi@raspberrypi:~/20200720/v5.5.1 $
```

到这个里面来进行安装

```
pi@raspberrypi:~/20200720/v5.5.1 $ sudo apt-get install raspberrypi-kernel-headers
Reading package lists... Done
Building dependency tree
Reading state information... Done
raspberrypi-kernel-headers is already the newest version (1.20201201-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~/20200720/v5.5.1 $
```

这个是安装 kernel 的。必须要安装的

然后配置下内核是 ARM

export ARCH=arm

安装 gcc 编译器，不然不能进行编译。

sudo apt-get install -y gcc make

```
pi@raspberrypi:~/20200720/v5.5.1 $ sudo apt-get install -y gcc make
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:8.3.0-1+rpi2).
make is already the newest version (4.2.1-1.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Make 编译下驱动文件

```
pi@raspberrypi:~/20200720/v5.5.1 $ make
rm -rf *~ .tmp_versions modules.order Module.symvers
find . -type f -name *~ -o -name *.o -o -name *.ko -o -name *.cmd -o -name *.mod.c | xargs rm -rf
make -C /lib/modules/5.4.79-v7l+/build M=/home/pi/20200720/v5.5.1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.79-v7l+'
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.o
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/usb_wwan.o
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/qcserial.o
Building modules, stage 2.
MODPOST 3 modules
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.mod.o
LD [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.ko
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/qcserial.mod.o
LD [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/qcserial.ko
```

sudo make install

```
LD [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/usb_wwan.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.79-v7l+'
pi@raspberrypi:~/20200720/v5.5.1 $ sudo make install
rm -rf *~ .tmp_versions modules.order Module.symvers
find . -type f -name *~ -o -name *.o -o -name *.ko -o -name *.cmd -o -name *.mod.c | xargs rm -rf
make -C /lib/modules/5.4.79-v7l+/build M=/home/pi/20200720/v5.5.1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.79-v7l+'
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.o
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/usb_wwan.o
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/qcserial.o
Building modules, stage 2.
MODPOST 3 modules
CC [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.mod.o
LD [M] /home/pi/20200720/v5.5.1/./drivers/usb/serial/option.ko
```

下面的是安装驱动，将驱动安装到系统里面去。

```
pi@raspberrypi:~/20200720/v5.5.1 $ lsusb -t
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 5000M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
|__ Port 3: Dev 5, If 0, Class=Communications, Driver=cdc_ether, 480M
|__ Port 3: Dev 5, If 1, Class=CDC Data, Driver=cdc_ether, 480M
|__ Port 3: Dev 5, If 2, Class=Vendor Specific Class, Driver=, 480M
|__ Port 3: Dev 5, If 3, Class=Vendor Specific Class, Driver=, 480M
|__ Port 3: Dev 5, If 4, Class=Vendor Specific Class, Driver=, 480M
pi@raspberrypi:~/20200720/v5.5.1 $
```

我们查 Driver=空，代表是没有驱动的。这样我们把树莓派重启看下

minicom -D /dev/ttyUSB1

设置拨号，非常简单

AT+CGDCONT=1,"IP","cmnet"

AT+QNETDEVCTL=1,1,1//配置网卡功能

Ifconfig

```
usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.100 netmask 255.255.255.0 broadcast 192.168.43.255
    inet6 fe80::d93f:c7f:1082:5483 prefixlen 64 scopeid 0x20<link>
    ether ae:0c:29:a3:9b:6d txqueuelen 1000 (Ethernet)
    RX packets 21 bytes 1766 (1.7 KiB)
    RX errors 1480 dropped 0 overruns 0 frame 0
    TX packets 125 bytes 13604 (13.2 KiB)
```

有 USB0 的相关信息，USB0 代表网卡接口

```
pi@raspberrypi:~$ ping www.baidu.com -I usb0
PING www.a.shifen.com (36.152.44.96) from 192.168.43.100 usb0: 56(84) bytes of data.
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=1 ttl=55 time=24.1 ms
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=2 ttl=55 time=21.2 ms
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=3 ttl=55 time=19.8 ms
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=4 ttl=55 time=18.8 ms
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=5 ttl=55 time=21.8 ms
64 bytes from 36.152.44.96 (36.152.44.96): icmp_seq=6 ttl=55 time=23.7 ms
```

可以看到 ping 通了。这样就可以使用 EC600S 作为网卡来给树莓派上网了